

A Note on the Evaluation of Mutation Prioritization Algorithms

Dusan Popovic^{*†}, Jesse Davis[‡], Alejandro Sifrim[§] and Bart De Moor^{*†}

^{*}STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
Department of Electrical Engineering (ESAT), KU Leuven
Kasteelpark Arenberg 10 - box 2446, B-3001 Leuven, Belgium

[†]iMinds Medical IT Department, KU Leuven
Kasteelpark Arenberg 10 - box 2446, B-3001 Leuven, Belgium

[‡]Department of Computer Science, KU Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

[§]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus
CB10 1SA, Hinxton - Cambridge, United Kingdom

Abstract—Recent developments in the field of gene sequencing technology greatly accelerated discovery of mutations that cause various genetic disorders. At the same time, a typical sequencing experiment generates a large number of candidate mutations, hence detecting single or few causative variants is still a formidable problem. Many computational methods have been proposed to assist this process, from which a large portion employ statistical learning in some form. Consequently, each newly designed algorithm is routinely compared to other competing systems in hope to demonstrate advantageous performance. In this work we review and discuss several issues related to the current practice of evaluation of mutation prioritization algorithms and suggest possible directions for improvements.

I. INTRODUCTION

The advent of high-throughput technologies, such as the next generation sequencing, has significantly accelerated biomedical research by enabling large portions of human genome to be simultaneously scanned. This ability facilitates the discovery of various alterations at a molecular level and has made high-throughput technologies an irreplaceable tool for studying complex [1], [2] and Mendelian diseases [3].

A typical sequencing study would compare the genomes of patients affected by certain condition to genomes of healthy individuals (e.g. unaffected family members, healthy population cohorts) to detect genes carrying a burden of pathogenic mutations [4]–[6]. However, each human genome harbors approximately 3.7 million single nucleotide variants (SNVs) [7] of which the vast majority are putatively neutral (i.e., they do not alter the fitness of their carrier) [8]. Given the prohibitive costs of confirmatory experiments (e.g. engineering CRISPR/CAS9 animal models), it is impossible to functionally validate all detected variants.

Moreover, given our current poor functional understanding of non-protein coding regions, it is often cost-effective to limit

the experiment to the protein coding regions of the genome (i.e. the exome). Limiting the scope to the exome reduces the number of mutations found by roughly 3 orders of magnitude (~ 20000 coding variants per exome). By applying further filtering and only retaining rare truncating loss-of-function mutations and amino-acid altering mutations (nSNVs) we can further reduce the search space for putatively pathogenic variation to hundreds of mutations (~ 500 rare protein-altering mutations per exome) [9]. Even these reduced numbers of mutations remains prohibitive for downstream studies and further prioritization is needed.

To address this challenge, a number of computational methods have been proposed to help discover disease-causing mutations. Many of these methods use biochemical, evolutionary or structural properties of the mutations under study to calculate a score that reflects their potential deleteriousness [10]–[16]. Some approaches combine multiple scores to obtain more reliable estimates [17], [18], while others are part of wider mutation prioritization frameworks that include various filtering steps and other features [19], [20]. Finally, several recent methods incorporate information about phenotypic representations of a disease to increase the precision of a detection [21], [22].

Many mutation prioritization algorithms employ learning, either directly or indirectly. For example, eXtasy [21] uses Random Forests [23] to model the complex relationship between several phenotypic and genomic features and the disease-causing potential of a mutation. In contrast, SIFT [10] takes a more indirect approach and provides an alignment-based score, thus generates a “training set” for each new prediction. Another example of indirect learning is the functional impact score for amino acid residue changes from Mutation Assessor [16] that is based on evolutionary conservation patterns.

Regardless of whether learning is employed directly or not, prioritization frameworks are evaluated on appropriate use-cases in order to quantify their performance. This is a crucial step, as it provides indication for their suitability in practice. Despite its importance, the impact of the evaluation methodology is less well understood and there is no accepted standard for evaluation. This paper reviews the current practice of evaluation of computational methods for mutation prioritization along several lines.

First, we discuss the choice of validation data set in terms of its domain, composition and class distribution. We identify limitations that are implicitly present in studies that use certain types of evaluation data and propose modifications that can help in overcoming these limitations.

Second, we analyze how various performance metrics are currently used to benchmark approaches and advocate a slight change of a perspective in this regard. Finally, we conclude the overall discussion with a short overview of lessons-learned and we outline possible directions of the further work.

II. EVALUATION DATA

When benchmarking prioritization systems, usually the goal is to evaluate how well each system distinguishes between mutations of interest (i.e., positive examples) and variants that are not interesting (i.e., negative examples). While several characteristics of the data are important in this regard, we will focus on two aspects: (1) what constitutes a positive and negative example, and (2) what is the ratio of positive to negative examples.

A. Domain of a testing set

The umbrella term *mutation prioritization* in reality covers wide variety of different algorithms that are designed with different goals in mind. Some systems are trained to distinguish neutral from deleterious variants, such as PolyPhen [24] or CADD [20]. Deleterious variants affect function of a gene but do not necessarily cause a genetic disorder, even sometimes one outcome is used as a proxy for another, either in training or evaluation (ex. SIFT [10]). In contrast, some methods are directly designed to distinguish disease-causing from neutral mutations (ex. CAROL score [17]). Some algorithms are further specialized in particular disease classes, such as rare genetic disorders [21], [22] or cancer [16] (see Table I).

Mutation prioritization algorithms also differ in *scope*. Some of them can process only one type of mutations, while some are capable of dealing with various types. For example, eXtasy works only on nSNVs [21], while Phen-Gen can analyze start-loss, stop-gain and stop-loss variants, small insertions and deletions in addition to nSNVs [22]. Furthermore, even the methods designed to address the same goal and that have the same scope are sometimes trained (or validated) on a fundamentally different composition of training data. For instance, the Phen-Gen model for nonsynonymous variants has been trained on data that include nonsynonymous substitutions in the human reference genome with respect to the ancestral sequence as controls, while eXtasy only uses rare neutral variants as negative examples.

The heterogeneity of validation data used for mutation prioritization gives rise to two types of evaluation problems.

First, algorithms designed for (slightly) different tasks are compared to each other. Second, sometimes the data used to validate an algorithm differs from the domain where the algorithm will be used in practice.

One example of the first type of validation issue is the common practice of comparing methods for discovering *disease-causing* mutations with methods that assess *deleteriousness* of variants. Among the algorithms that we reviewed here, Mutation Taster 2 [25], KGG-Seq [19] and eXtasy [21] are all tested against SIFT [10] and PolyPhen2 [11] on a data composed of disease-causing mutations and neutral variants. This practice is problematic, as deleteriousness of a variant does not automatically imply that a variant is disease-causing. In fact, it has been estimated that a genome of a healthy individual harbors up to one hundred variants that severely disrupt protein-coding genes [26].

An example of the second issue is the usage of common polymorphisms as controls when training and testing an algorithm for detection of disease-causing mutations, as in [16], [25]. In practice, common polymorphisms are usually filtered out before prioritization. Hence, when a validation data set is composed in this way, it is unclear if an evaluated method assesses the likelihood that a variant is disease-causing or it implicitly only assesses the *rarity* of a variant. As the rarity of a variant correlates with its likelihood of being involved in a disorder, this type of evaluation could lead to misleading results.

One of the basic assumptions of standard machine learning approaches is that examples that constitute training data and testing data are drawn independently from the same distribution [27], [28]. If this assumption is violated, an algorithm's performance may significantly decrease if applied to data drawn from a distribution that substantially deviates from that of a training data. Therefore, it is expected that methods tested on inadequate data will perform poorly, especially compared to the methods that have been tailored for use-case that this data represents.

We acknowledge that it is not always possible to find several algorithms that are designed to tackle *exactly the same* facet of the mutation prioritization problem as an evaluated method, which is the reason why often algorithms of slightly different but related function are used as a substitute. However, this limitation of study designs is not always clearly stated. Finally, researchers should strive to ensure that the validation data follows the distribution expected to be observed in practice.

B. Balance of a testing set

In addition to domain heterogeneity, evaluation data for mutation prioritization algorithms also differ in terms of class skew, which is the ratio of positive to negative examples. To illustrate this fact, Table I provides the class distribution for the data used to validate nine methods (SIFT [10], PolyPhen2 [11], PROVEAN [29], [30], CAROL [17], CONDEL [18], eXtasy [21], Mutation Assessor [16], Mutation Taster 2 [25] and VAAST [31]). Note that class skew varies significantly, even when only considering this small subset of prioritization algorithms. Concretely, for these data sets the class balance ranges from approximately one and a half positive instance

TABLE I. ALGORITHM’S TARGET AND CLASS BALANCE OF TESTING SETS USED FOR EVALUATION OF DIFFERENT MUTATION PRIORITIZATION ALGORITHMS

Method	Target variants	Skew of the testing set(s)
SIFT	deleterious	1.46, 0.51
PolyPhen2	deleterious	1.46, 0.50
PROVEAN	deleterious	0.86, 0.63, 0.25, 0.86
CAROL	deleterious	0.20
LRT	deleterious	real exomes
CONDEL	deleterious	1.50, 0.50
eXtasy	rare disease-causing	0.09, 0.56
Mutation Assessor	cancer	0.54
Mutation Taster 2	disease-causing	1.00
VAAST	disease-causing	1.00

per each negative instance (CONDEL), to eleven negative instances per each positive instance (eXtasy). In many cases, testing data contains approximately twice as much negatives than positives.

The first problem with the class distribution of the data typically used for evaluating mutation prioritization methods is that it barely ever corresponds to the true distribution in practice (even counterexamples exist, ex. LRT [13] is tested on real exomes). Usually, a class distribution is much less severe (i.e., closer to 1:1 ratio) in validation data than actual data. For instance, a single exome harbors several thousands of mutations, from which one or none is damaging. At the same time, methods that are designed for exome analysis are usually trained and tested on data with quite different balance [21].

Consequently, is very difficult for a potential user to anticipate how the algorithm will behave in practice, especially if the expected class balance (i.e., prevalence) is not provided. Moreover, this is not just a problem from the standpoint of interpreting performance metrics, but also impacts model selection. That is, different learning methods might be more or less appropriate, given the severity of the class distribution skew [32], [33]. It is not hard to envision that in many cases methods tailored for *outline detection* might be more suitable for dealing with extreme imbalance than classical classification algorithms, yet they can be selected against during model selection if tested on relatively balanced data.

Second, many performance metrics commonly used for evaluating mutation prioritization algorithms are in fact sensitive to changes in the class distribution. As an example, consider accuracy. In principle, under two very different class balances, several tested classifiers can be ranked differently in terms of accuracy, depending on which class they are “specialized in”. On the extreme end, using the accuracy to compare two algorithms on a highly imbalanced data set can misleadingly indicate that a trivial classifier (i.e., one that assigns the same label to all tested instances) performs better than less “accurate” one, while at the same time it is useless for prediction.

Arguably, it is not always possible to obtain realistic data to test novel algorithms. Nevertheless, the correct class balance can be *simulated* if intended use-case for an algorithm has been precisely defined beforehand. For example, to evaluate exome-based prioritization a synthetic data can be created by injecting disease-causing variant in a healthy exome, as done

in [19], [22]. Alternatively, the class distribution of a data set can be artificially altered. That is, one class can be subsampled to reflect a class balance that is expected under realistic usage scenario, as done in [34]. Moreover, the later approach can be easily combined with bootstrapping [35] to stabilize the estimates of the performance metrics. Finally, if none of these solutions can be easily implemented, then at least expected prevalence should be provided such that class balance-sensitive performance metrics can be calculated.

III. PERFORMANCE METRICS

In order to accurately quantify a system’s performance, it is important to select appropriate evaluation metrics that provide insight into how the system will perform in practice. In this section we discuss currently dominant, *classification-oriented* selection of performance metrics and suggest how a slight change of perspective on the mutation prioritization problem can facilitate better insight in future behavior of evaluated algorithms.

A. Classification perspective

Most mutation prioritization algorithms are in fact classifiers *by construction*. They are designed and tuned to distinguish positive (deleterious, disease-causing) from negative (neutral) cases, so hence provide scores or rankings only as a *by-product* of the classification process. Due to how these scores are derived, they seldom represent proper probabilities. For example, CADD [20] combines 63 features (including predictions from other algorithms, such as SIFT and PolyPhen) using a support vector machine classifier (SVM [36]) with a linear kernel. The score that CADD assigns to a variant is in essence a combination of scaled distances from several SVM decision hyperplanes and therefore it is not a probability.

Consequently, typical classification performance metrics based on a confusion matrix are routinely reported in articles that describe novel methods. Figure 1 provides a non-exhaustive list of basic metrics that can be derived from a confusion matrix. To illustrate the *de-facto* evaluation culture, Table II lists the reported performance measures for several mutation prioritization methods, including SIFT (new version [37]), PolyPhen2 [11], PROVEAN [29], [30], CAROL [17], LRT [13], CONDEL [18], eXtasy [21], Mutation Assessor [16], Mutation Taster 2 [25] and VAAST [31].

As is apparent from the Table II, estimates of accuracy, sensitivity and specificity are most often provided. As an alternative to accuracy, some authors report balanced accuracy. In some cases, the false positive and false negative rates (FPR and FNR) accompany the sensitivity and specificity. Finally, the Matthews correlation coefficient (MCC) is also occasionally reported on.

Because accuracy is an aggregate performance measure, it is not very useful for describing performance of a prioritization algorithm. Furthermore, as discussed in subsection II.B., care needs to be taken when interpreting its value when there is an extreme class imbalance, which is always the case in practical disease-causing mutation discovery. In contrast, balanced accuracy (i.e., mean of sensitivity and specificity) does not suffer from this problem, but it assumes that it is

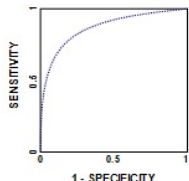
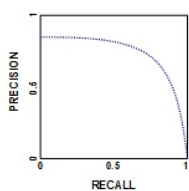
Confusion matrix :		$\text{Sens} = \frac{TP}{TP+FN}$	Sensitivity (or true positive rate (TPR), recall) is the probability of classifying real positive example as positive. False negative rate (FNR) is defined as $1 - \text{Sens}$.
Predicted		$\text{Spec} = \frac{TN}{TN+FP}$	Specificity (or true negative rate (TNR)) is the probability of classifying real negative example as negative. False positive rate (FPR) is defined as $1 - \text{Spec}$.
		$\text{Prec} = \frac{TP}{TP+FP}$	Precision (or positive predictive value (PPV)) is the probability that an example classified as positive is truly positive.
		$\text{NPV} = \frac{TN}{TN+FN}$	Negative predictive value (or negative precision) is the probability that an example classified as negative is truly negative.
		$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}$	Accuracy is a proportion of correctly predicted labels.
P – Positives N – Negatives TP – True positives TN – True negatives FP – False positives FN – False negatives		$\text{bAcc} = \frac{\text{Sens} + \text{Spec}}{2}$	Balanced accuracy is an average accuracy per class.
Graphical measures derived from confusion matrix			Receiver operating characteristic (ROC) curve displays Sensitivity/Specificity trade-off under varying classification thresholds.
			Precision-recall (PR) curve displays Precision/Recall trade-off under varying classification thresholds.

Fig. 1. Simple performance measures based on confusion matrix

TABLE II. PERFORMANCE METRICS REPORTED FOR DIFFERENT MUTATION PRIORITIZATION ALGORITHMS

Method	Performance metrics
SIFT	Acc, Sens, Spec, Prec, NPV, MCC, ROC
PolyPhen2	Sens for fixed FPR values (0.10,0.15,...,0.8), ROC
PROVEAN	bAcc, Sens, Spec, ROC, aROC
CAROL	Acc, Sens, Spec, FPR, FNR, ROC, aROC
LRT	Confusion matrix
CONDEL	Acc, ROC
eXtasy	Acc, Sens, Spec, Prec, NPV, MCC, ROC, aROC, PR, aPR
Mutation Assessor	Acc, ROC, aROC, distributions of scores
Mutation Taster 2	Acc, Sens, Spec, Prec, NPV, ROC
VAAST	Acc

Abbreviations : Acc (Accuracy), bAcc (Balanced accuracy), Sens (Sensitivity), Spec (Specificity), Prec (Precision), NPV (Negative predictive value), MCC (Matthews correlation coefficient), ROC (Receiver operating characteristic curve), aROC (area under the ROC curve), FPR (False positive rate), FNR (False negative rate), PR (Precision-recall curve), aPR (area under the PR curve)

equally important to correctly classify both positive and negative instances. Finally, FPR and FNR are just complements of sensitivity and specificity, so they do not provide an additional information about an algorithm's performance if sensitivity and specificity are given.

Receiver operating characteristic (ROC) curves, which show how an algorithm performances under different operating conditions, are also frequently provided. Surprisingly, the area under the ROC curve (aROC) is not always reported together with the graphical representation of the curve itself. The area under the ROC curve is proportional to Wilcoxon statistics,

hence to probability of producing correct pairwise ranking [38]. In other words, the aROC represents probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example. This equivalence allows estimating the *average rank* of positive instances, so it is strongly advisable to include it with the ROC curve.

However, even the aROC should be interpreted with caution. First, it may be misleading to compare two algorithms on the basis of this value alone as it is an aggregate measure of different operating conditions. The graphic representation is needed to ascertain under which operating ranges one classifier outperforms another. For example, if two ROC curves cross then it is possible that one curve has a larger value of aROC even though the alternative may have much better performance in the most important operating range. Second, aROC implicitly assumes a different misclassification cost distribution for each tested classifier, where this distribution directly depend on scores provided by the classifier [39].

Hence, from the metrics discussed so far, it is useful to provide sensitivity, specificity and the ROC curve together with exact aROC when reporting the performance of a mutation prioritization method. These measures provide information on proportions of positive and negative examples that can be captured by the algorithm for different decision thresholds. Matthews correlation coefficient is performance metric that is a bit less effective for this given purpose, as it does not have simple and direct interpretation. That is, MCC is in essence Pearson product-moment correlation coefficient calculated using the confusion matrix [40]. Nevertheless, all the measures

mentioned so far do not provide a complete information on prioritization performance, for the reasons that we exemplify in the next section.

B. Information retrieval perspective

In a typical application of a mutation prioritization algorithm one rarely conducts confirmatory experiments on *all* mutations that are classified as positive due to the high cost associated with the experimental verification. Therefore analyses are more often performed starting from the top of prioritized list of mutations, going downwards until genuine causative variant is found. From this perspective mutation prioritization is an instance of the information retrieval problem, rather than a standard classification task.

Hence, conventional classification performance metrics, such as sensitivity and specificity, do not capture all the important behavior characteristics of an algorithm. For completeness, this information should be supplemented with retrieval metrics, such as precision and PR curve [41], [42]. To further illustrate this issue we have constructed an artificial example that is displayed in Figure 2. This example has been conveniently created to emphasize the problem, yet it shares many typical features of real mutation prioritization problems. In fact, these features, such as the severe class imbalance, are even more pronounced in realistic use-cases than here.

The panel A in Figure 2 displays a number of (neutral and causative) mutations in a space that is spanned by the two predictive features, together with a hypothetical classification function. This classifier achieves decent sensitivity and specificity (0.7 and 0.78, respectively), but it is practically useless for prioritization due to a very low precision (0.11). The precision value implies that on average nine experiments have to be performed to find one causative variant. In contrast, using the classifier with a different learning bias (panel B) reduces this number to approximately four experiments, which is not readily apparent from its sensitivity and specificity, which are somewhat similar to the first approach (0.7 and 0.92, respectively). Moreover, the precision can not be even calculated from sensitivity and specificity alone: it requires knowing the prevalence.

The situation is further complicated when a ranking is introduced. Panels C and D on Figure 2 depict multiple decision boundaries for the two classifiers corresponding to various scores (i.e., decision thresholds). Each decision boundary produces different values of sensitivity, specificity and precision, which consequently results in a different point in ROC and PR spaces (see panels E and F). Obviously, both ROC and PR curves produced by the “square” classifier dominate over that of “circle” classifier. However, examining ROC curves alone might suggest that there is little difference between the two methods.

In contrast, PR curves emphasize this difference, especially in the vicinity of the highest ranks. For example, from the PR curve of the “square” classifier one can easily read that in the limit one experiment is needed to find the first causative mutation, five experiments to find the first three variants and so on. In contrast, this information is not readily apparent in the ROC curve. Hence, even though the dominance of one classifier over another in ROC space implies dominance

in PR space and vice versa [43], the size of the effect can significantly differ. In addition, PR curve has much more natural interpretation for prioritization tasks, as it clearly shows density of positives on the top of the prioritized list. However, as an aggregate measure, the area under the PR curve is less interpretable than aROC, due to unachievable regions in PR space which size depends on class skew [44].

Only a few of manuscripts enumerated in Table II include figures on precision, while even a smaller number show the full PR curve. However, in contrast to sensitivity and specificity (and consequently ROC curve) the precision (hence also PR curve) is class balance sensitive. Therefore, reporting on these metrics does not utilize their full descriptive potential if the class balance of a testing set do not correspond to a class balance that is expected in realistic use-case scenario, as discussed in section II.B.

Finally, the argument that we made about model selection in subsection II.B applies to the choice of performance metrics even more. That is, using a measure that is not suitable for assessing prioritization performance for model selection might lead to picking a suboptimal models. For example, if a model is selected as the best among several tested models according to its aROC, it might not be necessary better than other tested models in terms of aPR [43].

IV. CONCLUSION

We reviewed a portion of existing mutation prioritization algorithms to examine the current culture of predictive performance evaluation in this subfield of bioinformatics. We discussed several issues related to *ad-hoc* evaluation practice, including the choice of a testing set and performance metrics. In addition, we proposed various extensions of common validation procedures that can help to mitigate the identified problems.

We pointed out the great heterogeneity in the data sets used for the evaluation of variant prioritization methods and indicated where caution in interpreting the validation outcomes should be exercised and why. Consequently, we strongly believe that establishing a public repository, such as the UCI database for machine learning [45], would facilitate comparison of various methods and at the same time would greatly improve consistency of benchmarking results. An noteworthy example of work toward this goal is VariBench [46], a collection of data sets of experimentally verified variation data.

In parallel, some individual efforts towards independent assessment of mutation prioritization systems have been already undertaken (c.g., [47]). Nevertheless, we hope that community-wide initiatives for massive prospective evaluation of competing algorithms will also take place in the near future, as it is currently the case for protein function prediction algorithms [48].

Furthermore, we argued that mutation prioritization is in essence an information retrieval problem, and therefore we suggested to supplement classical classification evaluation metrics with their information retrieval counterparts when evaluating novel algorithms. However, as the values of most of these measures depend on the class distribution of the validation set, it is essential to apply them on data where the class balance matches the class balance expected in practice.

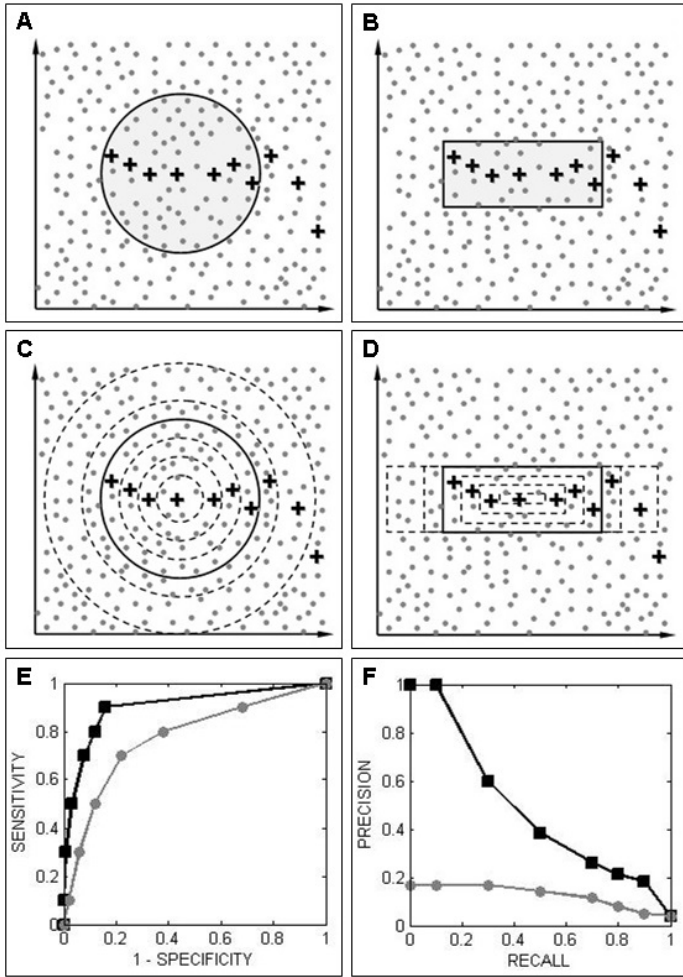


Fig. 2. **Hypothetical mutation prioritization problem and two classifiers used on it.** Gray dots represent negative, while "+" sign represent positive examples. Panels A and B depict two different classification functions (solid lines). Panels C and D depict decision boundaries corresponding to different thresholds of the two classifiers (dotted lines). Panels E and F depict ROC and PR curves for the two classifiers, respectively. Solid black lines with squares represent classifier figuring on the panel B, while gray lines with circles represent the classifier figuring on the panel A.

In this work, we restricted the discussion on the evaluation issues associated with testing data and performance metrics. However, other sources of potential validation problems can be identified in the literature on mutation prioritization. For example, not many evaluation pipelines employ statistical testing to prove significance of differences in performance between algorithms, even this is common practice in other fields and appropriate guidelines have been developed [49], [50]. Another example is the optimistic bias in performance estimates that can result from inadequate splitting a data set into disjoint training and testing partitions in cases when predictive features are defined on different levels of a hierarchy [34], [51].

Therefore, in the future we plan to conduct a comprehensive review of testing procedures used for assessing the performance of state-of-the-art mutation prioritization algorithms, as well as to perform a number of simulation experiments to better expose validation issues discussed here and other

potential problems. As a result, we hope to formalize a complete and consistent validation framework for this type of studies and to propose it to the research community.

ACKNOWLEDGMENT

The authors would like to acknowledge financial support from :

- Flemish Government:
 - FWO: projects: G.0871.12N (Neural circuits)
 - IWT : TBM-Logic Insulin(100793), TBM Rectal Cancer(100783), TBM IETA(130256); PhD grants
 - Industrial Research fund (IOF): IOF Fellowship 13-0260
 - iMinds Medical Information Technologies SBO 2015, ICON projects (MSIpad, My-HealthData)
 - VLK Stichting E. van der Schueren: rectal cancer
- Federal Government: FOD: Cancer Plan 2012-2015 KPC-29-023 (prostate)
- COST: Action: BM1104: Mass Spectrometry Imaging

Jesse Davis is partially supported by the Research Fund KU Leuven (OT/11/051), EU FP7 Marie Curie Career Integration Grant (#294068) and FWO-Vlaanderen (G.0356.12).

REFERENCES

- [1] S. Idris, S. Ahmad, M. Scott, G. Vassiliou, and J. Hadfield, "The role of high-throughput technologies in clinical cancer genomics," *Expert review of molecular diagnostics*, vol. 13, no. 2, p. 167, 2013.
- [2] M. Tuna and C. I. Amos, "Genomic sequencing in cancer," *Cancer letters*, vol. 340, no. 2, pp. 161–170, 2013.
- [3] J. J. McCarthy, H. L. McLeod, and G. S. Ginsburg, "Genomic medicine: a decade of successes, challenges, and opportunities," *Science translational medicine*, vol. 5, no. 189, pp. 189sr4–189sr4, 2013.
- [4] J. K. Van Houdt, B. A. Nowakowska, S. B. Sousa, B. D. van Schaik, E. Seuntjens, N. Avonce, A. Sifrim, O. A. Abdul-Rahman, M.-J. H. van den Boogaard, A. Bottani *et al.*, "Heterozygous missense mutations in smarca2 cause nicolaides-baraitser syndrome," *Nature genetics*, vol. 44, no. 4, pp. 445–449, 2012.
- [5] T. D. D. Study *et al.*, "Large-scale discovery of novel genetic causes of developmental disorders," *Nature*, 2014.
- [6] M. J. Bamshad, S. B. Ng, A. W. Bigham, H. K. Tabor, M. J. Emond, D. A. Nickerson, and J. Shendure, "Exome sequencing as a tool for mendelian disease gene discovery," *Nature Reviews Genetics*, vol. 12, no. 11, pp. 745–755, 2011.
- [7] H. Y. Lam, M. J. Clark, R. Chen, R. Chen, G. Natsoulis, M. O'Huallachain, F. E. Dewey, L. Habegger, E. A. Ashley, M. B. Gerstein *et al.*, "Performance comparison of whole-genome sequencing platforms," *Nature biotechnology*, vol. 30, no. 1, pp. 78–82, 2012.
- [8] J. Majewski, J. Schwartzentruber, E. Lalonde, A. Montpetit, and N. Jhabdo, "What can exome sequencing do for you?" *Journal of medical genetics*, vol. 48, no. 9, pp. 580–589, 2011.
- [9] J. R. Lupski, J. G. Reid, C. Gonzaga-Jauregui, D. Rio Deiros, D. C. Chen, L. Nazareth, M. Bainbridge, H. Dinh, C. Jing, D. A. Wheeler *et al.*, "Whole-genome sequencing in a patient with charcot-marie-tooth neuropathy," *New England Journal of Medicine*, vol. 362, no. 13, pp. 1181–1191, 2010.
- [10] P. C. Ng and S. Henikoff, "Sift: Predicting amino acid changes that affect protein function," *Nucleic acids research*, vol. 31, no. 13, pp. 3812–3814, 2003.

- [11] I. A. Adzhubei, S. Schmidt, L. Peshkin, V. E. Ramensky, A. Gerasimova, P. Bork, A. S. Kondrashov, and S. R. Sunyaev, "A method and server for predicting damaging missense mutations," *Nature methods*, vol. 7, no. 4, pp. 248–249, 2010.
- [12] J. M. Schwarz, C. Rödelberger, M. Schuelke, and D. Seelow, "Mutationtaster evaluates disease-causing potential of sequence alterations," *Nature methods*, vol. 7, no. 8, pp. 575–576, 2010.
- [13] S. Chun and J. C. Fay, "Identification of deleterious mutations within three human genomes," *Genome research*, vol. 19, no. 9, pp. 1553–1561, 2009.
- [14] T. Preeprem and G. Gibson, "Sds, a structural disruption score for assessment of missense variant deleteriousness," *Frontiers in genetics*, vol. 5, 2014.
- [15] S. Kumar, M. Sanderford, V. E. Gray, J. Ye, and L. Liu, "Evolutionary diagnosis method for variants in personal exomes," *Nature methods*, vol. 9, no. 9, pp. 855–856, 2012.
- [16] B. Reva, Y. Antipin, and C. Sander, "Predicting the functional impact of protein mutations: application to cancer genomics," *Nucleic acids research*, p. gkr407, 2011.
- [17] M. C. Lopes, C. Joyce, G. R. Ritchie, S. L. John, F. Cunningham, J. Asimit, and E. Zeggini, "A combined functional annotation score for non-synonymous variants," *Human heredity*, vol. 73, no. 1, pp. 47–51, 2012.
- [18] A. González-Pérez and N. López-Bigas, "Improving the assessment of the outcome of nonsynonymous snvs with a consensus deleteriousness score, condel," *The American Journal of Human Genetics*, vol. 88, no. 4, pp. 440–449, 2011.
- [19] M.-X. Li, H.-S. Gui, J. S. Kwan, S.-Y. Bao, and P. C. Sham, "A comprehensive framework for prioritizing variants in exome sequencing studies of mendelian diseases," *Nucleic acids research*, p. gkr1257, 2012.
- [20] M. Kircher, D. M. Witten, P. Jain, B. J. O’Roak, G. M. Cooper, and J. Shendure, "A general framework for estimating the relative pathogenicity of human genetic variants," *Nature genetics*, vol. 46, no. 3, pp. 310–315, 2014.
- [21] A. Sifrim, D. Popovic, L.-C. Tranchevent, A. Ardeschirdavani, R. Sakai, P. Konings, J. R. Vermeesch, J. Aerts, B. De Moor, and Y. Moreau, "extasy: variant prioritization by genomic data fusion," *Nature methods*, vol. 10, no. 11, pp. 1083–1084, 2013.
- [22] A. Javed, S. Agrawal, and P. C. Ng, "Phen-gen: combining phenotype and genotype to analyze rare disorders," *Nature methods*, vol. 11, no. 9, pp. 935–937, 2014.
- [23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.
- [24] V. Ramensky, P. Bork, and S. Sunyaev, "Human non-synonymous snps: server and survey," *Nucleic acids research*, vol. 30, no. 17, pp. 3894–3900, 2002.
- [25] J. M. Schwarz, D. N. Cooper, M. Schuelke, and D. Seelow, "Mutationtaster2: mutation prediction for the deep-sequencing age," *Nature methods*, vol. 11, no. 4, pp. 361–362, 2014.
- [26] D. G. MacArthur, S. Balasubramanian, A. Frankish, N. Huang, J. Morris, K. Walter, L. Jostins, L. Habegger, J. K. Pickrell, S. B. Montgomery *et al.*, "A systematic survey of loss-of-function variants in human protein-coding genes," *Science*, vol. 335, no. 6070, pp. 823–828, 2012.
- [27] L. G. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.
- [28] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.
- [29] Y. Choi, G. E. Sims, S. Murphy, J. R. Miller, and A. P. Chan, "Predicting the functional effect of amino acid substitutions and indels," *PLoS ONE*, vol. 7, no. 10, 2012.
- [30] Y. Choi and A. P. Chan, "Provean web server: a tool to predict the functional effect of amino acid substitutions and indels," *Bioinformatics*, p. btv195, 2015.
- [31] M. Yandell, C. Huff, H. Hu, M. Singleton, B. Moore, J. Xing, L. B. Jorde, and M. G. Reese, "A probabilistic disease-gene finder for personal genomes," *Genome research*, vol. 21, no. 9, pp. 1529–1542, 2011.
- [32] S. Daskalaki, I. Kopanas, and N. Avouris, "Evaluation of classifiers for an uneven class distribution problem," *Applied artificial intelligence*, vol. 20, no. 5, pp. 381–417, 2006.
- [33] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.
- [34] D. Popovic, A. Sifrim, J. Davis, Y. Moreau, and B. De Moor, "Problems with the nested granularity of feature domains in bioinformatics: the extasy case," *BMC bioinformatics*, vol. 16, no. Suppl 4, p. S2, 2015.
- [35] B. Efron, "Bootstrap methods: another look at the jackknife," *The annals of Statistics*, pp. 1–26, 1979.
- [36] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2013.
- [37] N.-L. Sim, P. Kumar, J. Hu, S. Henikoff, G. Schneider, and P. C. Ng, "Sift web server: predicting effects of amino acid substitutions on proteins," *Nucleic acids research*, vol. 40, no. W1, pp. W452–W457, 2012.
- [38] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [39] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine learning*, vol. 77, no. 1, pp. 103–123, 2009.
- [40] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.
- [41] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.
- [42] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.
- [43] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 233–240.
- [44] K. Boyd, V. S. Costa, J. Davis, and C. D. Page, "Unachievable region in precision-recall space and its effect on empirical evaluation," in *Machine learning: proceedings of the International Conference. International Conference on Machine Learning*, vol. 2012. NIH Public Access, 2012, p. 349.
- [45] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.
- [46] P. S. Nair and M. Vihinen, "VariBench: a benchmark database for variations," *Human mutation*, vol. 34, no. 1, pp. 42–49, 2013.
- [47] C. Dong, P. Wei, X. Jian, R. Gibbs, E. Boerwinkle, K. Wang, and X. Liu, "Comparison and integration of deleteriousness prediction methods for nonsynonymous snvs in whole exome sequencing studies," *Human molecular genetics*, vol. 24, no. 8, pp. 2125–2137, 2015.
- [48] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim, C. Funk, K. Verspoor, A. Ben-Hur *et al.*, "A large-scale evaluation of computational protein function prediction," *Nature methods*, vol. 10, no. 3, pp. 221–227, 2013.
- [49] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [50] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1993, pp. 329–338.
- [51] D. Popovic, A. Sifrim, J. Davis, Y. Moreau, and B. De Moor, "Improving performance of the extasy model by hierarchical sampling," in *Pattern Recognition in Bioinformatics: 9th IAPR International Conference, PRIB 2014, Stockholm, Sweden, August 21-23, 2014. Proceedings*, vol. 8626. Springer, 2014, pp. 125–128.